Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence

Accurate Probability Calibration for Multiple Classifiers

Leon Wenliang Zhong **James T. Kwok**
Department of Computer Science and Engineering
Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong
{wzhong, jamesk}@cse.ust.hk

Abstract

In classification problems, isotonic regression has been commonly used to map the prediction scores to posterior class probabilities. However, isotonic regression may suffer from overfitting, and the learned mapping is often discontinuous. Besides, current efforts mainly focus on the calibration of a single classifier. As different classifiers have different strengths, a combination of them can lead to better performance. In this paper, we propose a novel probability calibration approach for such an ensemble of classifiers. We first construct isotonic constraints on the desired probabilities based on soft voting of the classifiers. Manifold information is also incorporated to combat overfitting and ensure function smoothness. Computationally, the extended isotonic regression model can be learned efficiently by a novel optimization algorithm based on the alternating direction method of multipliers (ADMM). Experiments on a number of real-world data sets demonstrate that the proposed approach consistently outperforms independent classifiers and other combinations of the classifiers' probabilities in terms of the Brier score and AUC.

1 Introduction

In many classification problems, it is important to estimate the posterior probabilities that an instance belongs to each of the output classes. For example, in medical diagnosis, it is more natural to estimate the patient's probability of having cancer, rather than simply giving an assertion [Gail *et al.*, 1989]; in computational advertising, it is useful to estimate the probability that an advertisement will be clicked [Richardson *et al.*, 2007]. Moreover, different misclassifications may have different costs, which need not even be known during training. In order to make cost-sensitive decisions, probability is again an essential component in the computation of the conditional risk [Zadrozny and Elkan, 2001].

However, many popular classifiers, such as the SVM and boosting, can only output a prediction score; while others, such as the naive Bayes classifier, are unable to produce accurate probability estimates [Niculescu-Mizil and Caruana, 2005]. Calibration of these scores or probabilities is thus an important research issue. Currently, the most popular calibration methods are Platt scaling [Platt, 1999] and isotonic regression [Zadrozny and Elkan, 2001; 2002]. Platt scaling is based on fitting the scores with a sigmoid. This, however, may not be the right transformation for many classifiers [Niculescu-Mizil and Caruana, 2005]. Isotonic regression, on the other hand, is nonparametric and only needs to assume that the calibrated probability is monotonically increasing with the score. It has demonstrated great empirical success on various classifiers [Niculescu-Mizil and Caruana, 2005; Caruana and Niculescu-Mizil, 2006; Caruana *et al.*, 2008], and has also outperformed Platt's method on most problems [Caruana *et al.*, 2008]. Recently, this is further improved by generating the isotonic constraints based on a direct optimization of the AUC via ranking [Menon *et al.*, 2012].

Though flexible, isotonic regression can suffer from overfitting, especially with limited calibration data [Niculescu-Mizil and Caruana, 2005]. Moreover, as the construction of isotonic constraints depends only on the scores' ordering, similar scores need not yield similar calibrated probabilities. Indeed, the isotonic regression function is not even continuous in general and can have jumps (Figure 1). This is often undesirable and may hurt prediction performance. A variety of techniques have been proposed to smooth out the discontinuities, such as by using moving average [Friedman and Tibshirani, 1984], kernel estimator [Hall and Huang, 2001; Jiang *et al.*, 2011] and smoothing spline [Wang and Li, 2008]. However, they are applicable only when the isotonic constraints are ordered on a one-dimensional list.

Another limitation of existing calibration algorithms is that they only focus on one single classifier. As different classifiers may have different strengths, it is well-known that ensemble learning can improve performance [Zhou, 2012]. A standard ensemble approach is to average (possibly weighted) the probabilities obtained from all the classifiers. As will be seen in Section 4, empirically this can be outperformed by better approaches proposed in the following.

In this paper, we extend the isotonic regression model to alleviate the above problems. First, instead of constructing isotonic constraints individually for each classifier, we construct a more refined set of isotonic constraints based on the vector of scores obtained from all the classifiers. Moreover, to avoid overfitting and ensure smoothness of the calibrated probabili-

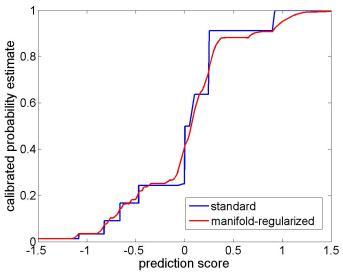


Figure 1: Calibration curve obtained on the *ijcnn1* data set. The abscissa is the classifier score and the ordinate is the calibrated probability produced by isotonic regression with (red) and without manifold regularization (blue).

ties with respect to the scores, we incorporate the highly successful technique of manifold regularization [Belkin *et al.*, 2006]. To learn this extended model, we propose a novel optimization algorithm based on the alternating direction method of multipliers (ADMM) [Boyd, 2010], which has attracted significant interest recently in diverse fields such as machine learning, data mining and image processing.

The rest of this paper is organized as follows. Section 2 first gives a brief review of isotonic regression and ADMM. Section 3 describes the proposed calibration model and its solver. Experimental results are presented in Section 4, and the last section gives some concluding remarks.

Notations: In the sequel, matrices and vectors are denoted in bold, with upper-case letters for matrices and lower-case for vectors. The transpose of a vector/matrix is denoted by the superscript $^\top$.

2 Related Work

2.1 Isotonic Regression for Probability Calibration

Isotonic regression has been used in diverse areas including physics, chemistry, biology, operations research, and statistics [Barlow *et al.*, 1972]. Given a set of observations $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$, isotonic regression finds the estimates $\{f_1, \ldots, f_n\}$ at the $\mathbf{x}_i$'s such that the model (i) fits the data with minimum error w.r.t. a convex loss function; and (ii) satisfies the isotonic constraints: $f_i \geq f_j$ if $\mathbf{x}_i \succeq \mathbf{x}_j$. Here, $\succeq$ is an application-specific partial order defined on the $\mathbf{x}_i$'s, and is often represented by a directed acyclic graph (DAG).[1] Over the decades, solvers have been developed for various combinations of loss functions (such as ℓ_1, ℓ_2 and ℓ_∞) and subclasses of DAG (such as general DAGs, trees, grids, and linear lists). A recent survey can be found in [Stout, 2013].

In the context of probability calibration for a single classifier [Zadrozny and Elkan, 2002], f_i is the calibrated probability of pattern i that is to be estimated, input x_i is the the classifier's prediction score, and output $y_i = 1$ if the pattern belongs to the positive class; and 0 otherwise. Since the x_i's are scalars here, the partial order "$\succeq$" becomes a total order, and can be easily obtained by sorting the x_i's. Intuitively, this amounts to assuming that the mapping from scores to probabilities is non-decreasing. With the commonly-used square loss, this isotonic regression problem can be formulated as:

$$\min_{f_1,\ldots,f_n} \sum_{i=1}^{n} (y_i - f_i)^2 \ : \ f_i \geq f_j \text{ if } x_i \geq x_j. \qquad (1)$$

[1] In the DAG representation, each vertex v_i corresponds to a $\mathbf{x}_i$, and there is a directed edge from v_i to v_j if $\mathbf{x}_i \succeq \mathbf{x}_j$.

2.2 Alternating Direction Method of Multipliers (ADMM)

ADMM is a simple but powerful algorithm first introduced in the 1970s [Glowinski and Marrocco, 1975], and has recently been popularly used in diverse fields such as machine learning, data mining and image processing [Boyd, 2010]. It can be used to solve optimization problems of the form

$$\min_{\mathbf{x},\mathbf{y}} \ \phi(\mathbf{x}) + \psi(\mathbf{y}) \ : \ \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y} = \mathbf{c},$$

where $\phi(\cdot), \psi(\cdot)$ are convex functions, and $\mathbf{A}, \mathbf{B}$ (resp. $\mathbf{c}$) are constant matrices (resp. vector) of appropriate sizes. As in the method of multipliers, ADMM considers the augmented Lagrangian: $L(\mathbf{x}, \mathbf{y}, \mathbf{u}) = \phi(\mathbf{x}) + \psi(\mathbf{y}) + \mathbf{u}^\top(\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y} - \mathbf{c}) + \frac{\rho}{2}\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y} - \mathbf{c}\|^2$, where $\mathbf{u}$ is the vector of Lagrangian multipliers, and $\rho > 0$ is a penalty parameter. At the kth iteration of ADMM, the values of $\mathbf{x}, \mathbf{y}$ and $\mathbf{u}$ (denoted $\mathbf{x}^k, \mathbf{y}^k$ and $\mathbf{u}^k$) are updated as

$$\begin{aligned} \mathbf{x}^{k+1} &= \arg\min_{\mathbf{x}} L(\mathbf{x}, \mathbf{y}^k, \mathbf{u}^k), \\ \mathbf{y}^{k+1} &= \arg\min_{\mathbf{y}} L(\mathbf{x}^{k+1}, \mathbf{y}, \mathbf{u}^k), \\ \mathbf{u}^{k+1} &= \mathbf{u}^k + \rho(\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{y}^{k+1} - \mathbf{c}). \end{aligned}$$

Note that ADMM minimizes $L(\mathbf{x}, \mathbf{y}, \mathbf{u}^k)$ w.r.t. $\mathbf{x}$ and $\mathbf{y}$ in an alternating manner, while the method of multipliers minimizes $\mathbf{x}$ and $\mathbf{y}$ jointly. This allows ADMM to more easily decompose the optimization problem when ϕ and ψ are separable. Let $\mathbf{r} = \frac{\mathbf{u}}{\rho}$ be the scaled dual variable, the ADMM procedure can be expressed as in Algorithm 1 [Boyd, 2010].

Algorithm 1 The ADMM algorithm.

1: Initialize $\mathbf{x}^0, \mathbf{y}^0, \mathbf{r}^0$, set $t \leftarrow 0$;
2: **repeat**
3: $\quad \mathbf{x}^{t+1} \leftarrow \arg\min_{\mathbf{x}} \phi(\mathbf{x}) + \frac{\rho}{2}\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y}^t - \mathbf{c} + \mathbf{r}^t\|^2$;
4: $\quad \mathbf{y}^{t+1} \leftarrow \arg\min_{\mathbf{y}} \psi(\mathbf{y}) + \frac{\rho}{2}\|\mathbf{A}\mathbf{x}^{t+1} + \mathbf{B}\mathbf{y} - \mathbf{c} + \mathbf{r}^t\|^2$;
5: $\quad \mathbf{r}^{t+1} \leftarrow \mathbf{r}^t + (\mathbf{A}\mathbf{x}^{t+1} + \mathbf{B}\mathbf{y}^{t+1} - \mathbf{c})$;
6: $\quad t \leftarrow t + 1$;
7: **until** convergence;
8: **return** $\mathbf{x}^t, \mathbf{y}^t$ obtained in the last iteration.

3 Calibration for Multiple Classifiers

Given a set of C classifiers (such as the SVM, logistic regressor, boosted decision trees, etc.), we propose to obtain a calibrated probability estimate by utilizing all C prediction scores. Section 3.1 presents an extension of the isotonic regression approach in [Zadrozny and Elkan, 2002]. Section 3.2 combats the overfitting and smoothness problems of isotonic regression by incorporating manifold regularization. Finally, Section 3.3 proposes an ADMM-based solver for the resultant optimization problem.

3.1 Construction of the Isotonic Constraints

For pattern i, let $\mathbf{x}_i = [x_{i1}, \ldots, x_{iC}]^\top$ be the vector of scores obtained from the C classifiers. Recall that $\{f_1, \ldots, f_n\}$ are the calibrated probabilities to be estimated, and that the mapping from scores to probabilities is assumed to be non-decreasing. A natural extension of [Zadrozny and Elkan, 2002] is to require $f_i \geq f_j$ if all C classifiers agree, i.e., $x_{ic} \geq x_{jc}$ for $c = 1, \ldots, C$. However, unless C is small, getting this consensus may be too stringent. This will be particularly problematic when some classifiers are not accurate.

To alleviate this problem, we perform soft voting of the classifiers. Specifically, different weights η_c's, where $\eta_c \geq 0$ and $\sum_{c=1}^{C} \eta_c = 1$, are assigned to the classifiers. An isotonic constraint $f_i \geq f_j$ is constructed if $\sum_{c=1}^{C} \eta_c I(x_{ic} \geq x_{jc}) \geq \alpha$, where $\alpha \in (0.5, 1]$ is a user-defined threshold, and $I(\cdot)$ is the indicator function which returns 1 when the argument holds, and 0 otherwise. Problem (1) is then modified as:

$$\begin{aligned} \min_{f_1,\ldots,f_n} \quad & \sum_{i=1}^{n} (y_i - f_i)^2 \qquad (2) \\ \text{s.t.} \quad & f_i \geq f_j \text{ if } \sum_{c=1}^{C} \eta_c I(x_{ic} \geq x_{jc}) \geq \alpha. \end{aligned}$$

In general, the isotonic constraints above may lead to a directed graph with cycles, as when $f_i \geq f_j \geq f_k \geq \cdots \geq f_i$. In this paper, we use topological sort to detect such cycles [Cormen *et al.*, 2009], and remove all the associated constraints. Problem (2) is then a standard isotonic regression problem with constraints ordered on a DAG $G(V, E)$, where V denotes the set of vertices and E is the set of edges.

3.2 Incorporating Manifold Regularization

To combat the overfitting and smoothness problems in isotonic regression, we encourage the regression outputs (i.e., calibrated probabilities) for patterns i and j to be close if their score vectors $\mathbf{x}_i, \mathbf{x}_j$ are similar. This can be implemented with the manifold regularizer $\sum_{e_{ij} \in E} \omega_{ij} (f_i - f_j)^2$. Here, ω_{ij} measures the similarity between $\mathbf{x}_i, \mathbf{x}_j$, and can be set by prior knowledge or as a function of the distance between $\mathbf{x}_i, \mathbf{x}_j$. Let $\mathbf{f} = [f_1, \ldots, f_n]^\top$. It is well-known that the manifold regularizer can be written as $\mathbf{f}^\top \mathbf{\Omega} \mathbf{f}$, where $\mathbf{\Omega}$ is the graph Laplacian matrix of G. Adding this to (2), we then have

$$\min_{\mathbf{f}} \sum_{i=1}^{n} (y_i - f_i)^2 + \frac{\lambda}{2} \mathbf{f}^\top \mathbf{\Omega} \mathbf{f} \ : \ f_i \geq f_j \text{ if } e_{ij} \in E, \quad (3)$$

where λ is a regularization parameter.

3.3 Optimization Solver for the Extended Model

Obviously, problem (3) reduces to standard isotonic regression when $\lambda = 0$. However, no existing solver can handle the case of $\lambda \neq 0$ on general DAG ordering. Moreover, while smoothing and spline regularization have been used with isotonic regression as reviewed in Section 1, they can only be used with constraints ordered on a one-dimensional list, but not on a DAG ordering as we have here.

In the sequel, we first convert the DAG ordering in (3) to an equivalent tree ordering with additional constraints, and then apply ADMM in Section 2.2. As will be seen, one of the ADMM update steps has a simple closed-form solution, while the other can be reduced to a standard isotonic regression problem on tree orderings.

Converting the DAG Ordering to a Tree Ordering

The conversion algorithm first checks the number of parents ($n_{\text{par}}(i)$) for every vertex $v_i \in V$. If $n_{\text{par}}(i) > 1$, we duplicate v_i ($n_{\text{par}}(i) - 1$) times and add edges such that each of its parents is connected to a copy of v_i, thus forming a tree[2] T (Figure 2). For any $\mathbf{f} \in \mathbb{R}^n$ defined on the nodes of G, the corresponding vector defined on the nodes of T is $\hat{\mathbf{f}} = [\hat{f}_{1,1}, \underbrace{\hat{f}_{2,1}, \hat{f}_{2,2}, \ldots, \hat{f}_{2,n_{\text{par}}(2)}}_{n_{\text{par}}(2) \text{ times}}, \ldots, \underbrace{\hat{f}_{n,1}, \hat{f}_{n,2}, \ldots, \hat{f}_{n,n_{\text{par}}(n)}}_{n_{\text{par}}(n) \text{ times}}]^\top \in \mathbb{R}^{|E|+1}$. Here, the root has index 1. For notational simplicity, we set $n_{\text{par}}(1) = 1$, and thus $\hat{f}_{1,1} = f_1$. By construction, if $\mathbf{f}$ satisfies the isotonic constraints in G, $\hat{\mathbf{f}}$ also satisfies the isotonic constraints in T. Moreover, it can be easily seen that $\hat{\mathbf{f}}$ and $\mathbf{f}$ are related as $\hat{\mathbf{f}} = \mathbf{Q}\mathbf{f}$, where $\mathbf{Q} \in \mathbb{R}^{(|E|+1) \times n}$ with rows indexed in the same order as $\hat{\mathbf{f}}$; and

$$Q_{ek} = 1 \text{ if } j = k; 0 \text{ otherwise}, \qquad (4)$$

where $e = e_{ij} \in E$ is an edge from v_i to v_j. Note also that $(f_i - y_i)^2 = \frac{1}{n_{\text{par}}(i)} \sum_{p=1}^{n_{\text{par}}(i)} (\hat{f}_{i,p} - y_i)^2$. Hence, problem (3) can be rewritten as

$$\begin{aligned} \min_{\hat{\mathbf{f}}, \mathbf{f}} \ & \underbrace{\delta(\hat{\mathbf{f}}) + \sum_{i=1}^{n} \frac{1}{n_{\text{par}}(i)} \sum_{p=1}^{n_{\text{par}}(i)} (\hat{f}_{i,p} - y_i)^2}_{\phi(\hat{\mathbf{f}})} + \underbrace{\frac{\lambda}{2} \mathbf{f}^\top \mathbf{\Omega} \mathbf{f}}_{\psi(\mathbf{f})} \qquad (5) \\ \text{s.t} \ & \hat{\mathbf{f}} = \mathbf{Q}\mathbf{f}, \end{aligned}$$

where $\delta(\hat{\mathbf{f}}) = 0$ if $\hat{\mathbf{f}}$ satisfies the isotonic constraints in T; and ∞ otherwise.

Using ADMM

By defining $\phi(\cdot)$ and $\psi(\cdot)$ as shown in (5), we now use the ADMM to obtain an ϵ-approximate solution of (5). Recall that ADMM involves two key steps: (i) the updating of $\hat{\mathbf{f}}$ (step 3 in Algorithm 1), and (ii) the updating of $\mathbf{f}$ (step 4). The first subproblem can be rewritten as

$$\begin{aligned} & \min_{\hat{\mathbf{f}}} \quad \phi(\hat{\mathbf{f}}) + \frac{\rho}{2} \|\hat{\mathbf{f}} - \mathbf{Q}\mathbf{f}^t + \mathbf{r}^t\|^2 \\ = \ & \min_{\hat{\mathbf{f}}} \quad \phi(\hat{\mathbf{f}}) + \frac{\rho}{2} \sum_{i=1}^{n} \sum_{p=1}^{n_{\text{par}}(i)} (\hat{f}_{i,p} - f_i^t + r_{i,p}^t)^2 \\ = \ & \min_{\hat{\mathbf{f}}} \quad \sum_{i,p} w_{i,p} (\hat{f}_{i,p} - c_{i,p})^2 \qquad (6) \\ & \text{s.t.} \quad \hat{\mathbf{f}} \text{ satisfies the isotonic constraints in } T, \end{aligned}$$

where

$$w_{i,p} = \frac{1}{n_{\text{par}}(i)} + \frac{\rho}{2}, \quad c_{i,p} = \frac{2y_i + \rho n_{\text{par}}(i)(f_i^t - r_{i,p}^t)}{\rho n_{\text{par}}(i) + 2},$$

[2] We assume that the DAG has a single root. Otherwise, a pseudo-root, with $f_{\text{root}} = 1$, is added and connected to all the original roots.

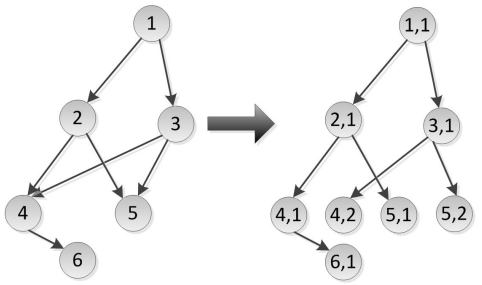


Figure 2: Converting a DAG to a tree.

and the last equality is obtained by completing squares with the quadratic term in $\phi(\hat{\mathbf{f}})$. Problem (6) is a standard isotonic regression problem on tree ordering, and can be solved efficiently in $O(|E|\log|E|)$ time [Pardalos and Xue, 1999].

For the second subproblem $\min_{\mathbf{f}} \frac{\lambda}{2}\mathbf{f}^\top\mathbf{\Omega}\mathbf{f} + \frac{\rho}{2}\|\hat{\mathbf{f}}^{t+1} - \mathbf{Q}\mathbf{f} + \mathbf{r}^t\|^2$, on setting the derivative of its objective to zero, the optimal $\mathbf{f}$ can be easily obtained as $\rho(\lambda\mathbf{\Omega}+\rho\mathbf{Q}^\top\mathbf{Q})^{-1}\mathbf{Q}^\top(\hat{\mathbf{f}}^{t+1}+\mathbf{r}^t)$. Note that $(\lambda\mathbf{\Omega}+\rho\mathbf{Q}^\top\mathbf{Q})^{-1}$ does not change throughout the iterations and so can be pre-computed.

To terminate ADMM, we require the primal residual $\|\hat{\mathbf{f}}^t - \mathbf{Q}\mathbf{f}^t\|$ and dual residual $\rho\|\mathbf{Q}\mathbf{f}^t - \mathbf{Q}\mathbf{f}^{t-1}\|$ are small [Boyd, 2010]. The complete procedure is shown in Algorithm 2. In the sequel, the formulation in (2) will be called Multi-Isotonic-regression-based Calibration (MIC), and its Manifold-Regularized extension in (3) MR-MIC.

Algorithm 2 Algorithm to solve the MR-MIC model in (3).

1: Convert problem (3) to problem (5);
2: $t \leftarrow 0$; set $\hat{\mathbf{f}}^0, \mathbf{f}^0, \mathbf{r}^0 \leftarrow \mathbf{0}$;
3: **repeat**
4: $\quad\hat{\mathbf{f}}^{t+1} \leftarrow$ solve (6) using standard isotonic regression solver;
5: $\quad\mathbf{f}^{t+1} \leftarrow \rho(\lambda\mathbf{\Omega}+\rho\mathbf{Q}^\top\mathbf{Q})^{-1}\mathbf{Q}^\top(\hat{\mathbf{f}}^{t+1}+\mathbf{r}^t)$;
6: $\quad\mathbf{r}^{t+1} \leftarrow \mathbf{r}^t + (\hat{\mathbf{f}}^{t+1} - \mathbf{Q}\mathbf{f}^{t+1})$;
7: $\quad t \leftarrow t+1$;
8: **until** convergence.
9: **return** $\mathbf{f}^t$ obtained in the last iteration.

Time Complexity

It is easy to see that converting the DAG to a tree in Section 3.3 takes $\mathcal{O}(|E|)$ time. As $\phi(\hat{\mathbf{f}})$ is strongly convex and $\mathbf{Q}$ is full rank, an ϵ-approximate solution of (5) can be obtained by ADMM in $\mathcal{O}(\log\frac{1}{\epsilon})$ iterations [Deng and Yin, 2012]. In each iteration, step 4 takes $\mathcal{O}(|E|\log|E|)$ time [Pardalos and Xue, 1999]. For step 5, note from (4) that $\mathbf{Q}$ is sparse and has only $\mathcal{O}(|E|)$ nonzero entries. Hence, computing $\mathbf{Q}^\top(\hat{\mathbf{f}}^{t+1}+\mathbf{r}^t)$ only takes $\mathcal{O}(|E|)$ time. Assuming that the $n\times n$ matrix inverse $(\lambda\mathbf{\Omega}+\rho\mathbf{Q}^\top\mathbf{Q})^{-1}$ has been pre-computed, step 5 then takes $\mathcal{O}(n^2+|E|) = \mathcal{O}(n^2)$ time.[3] Hence, Algorithm 2 takes a total of $\mathcal{O}\left(\log\frac{1}{\epsilon}(|E|\log|E|+n^2)\right)$ time. When G is simply a tree, $|E| = \mathcal{O}(n)$ and the total complexity reduces to $\mathcal{O}\left(\log\left(\frac{1}{\epsilon}\right)(n\log n + n^2)\right) = \mathcal{O}\left(\log\left(\frac{1}{\epsilon}\right)n^2\right)$.

[3] In case $(\lambda\mathbf{\Omega}+\rho\mathbf{Q}^\top\mathbf{Q})^{-1}$ cannot be stored, one can use its rank-k approximation and Step 5 then takes $\mathcal{O}(nk+|E|)$ time.

4 Experiments

In this section, experiments are performed on five standard binary classification data sets (*real-sim*, *news20*, *rcv1*, *ijcnn1*, and *covertype*) from the LIBSVM archive. Three of these (*real-sim*, *news20* and *rcv1*) are text data sets, *ijcnn1* comes from the IJCNN 2001 neural network competition, and *covertype* contains remote sensing image data. For each data set, 1,000 samples are used for training, 1,000 for validation, and another 10,000 samples for testing. As in [Niculescu-Mizil and Caruana, 2005], the validation set is used for both parameter tuning of the classifiers and training of the isotonic regression model. To reduce statistical variability, results are averaged over 10 repetitions.

In the experiment, we first train and calibrate a number of classifiers by isotonic regression [Zadrozny and Elkan, 2002]. The following approaches to combine the calibrated probabilities of classifiers will be compared:

1. avg: simple averaging of the calibrated probabilities;
2. wavg: weighted averaging of the calibrated probabilities based on the performance of the classifiers. Specifically, the weight of classifier c is defined as
$$\eta_c = \frac{1}{Z}\exp\left(\frac{-(1-\text{AUC}_c)}{2\mu}\right), \tag{7}$$
where AUC_c is the area under the ROC curve [Fawcett, 2006] obtained by classifier c on the validation set, μ is the average of $(1-\text{AUC}_c)$ over the C classifiers, and Z normalizes $\{\eta_c\}_{c=1}^C$ to sum to 1. Intuitively, the higher the classifier's AUC, the larger its weight.
3. MIC (model (2)): The isotonic constraints are constructed using the weights in (7), and with $\alpha = 0.8$.
4. MR-MIC (model (3)): The similarity between scores $\mathbf{x}_i, \mathbf{x}_j$ on the manifold is set as $\omega_{ij} = \frac{1}{\|\mathbf{x}_i - \mathbf{x}_j\|}$. Recall that the validation set becomes the training set for the isotonic model. We set aside $1/4$ of it to tune λ. Moreover, the penalty ρ in Algorithm 2 is simply set to 1. In practice, convergence can often be improved by dynamically adjusting its value [Boyd, 2010].

For performance evaluation, we use the following two criteria that are commonly used for probability calibration [Caruana *et al.*, 2008; Niculescu-Mizil and Caruana, 2005]:

1. mean square error (MSE): $\frac{1}{n}\sum_{i=1}^n(y_i - f_i)^2$, which is also called the Brier score [Brier, 1950]; and
2. area under the ROC curve (AUC) [Fawcett, 2006].

4.1 Combining Similar and Dissimilar Classifiers

We use three classifiers (i) linear SVM (SVM-lin) [Fan *et al.*, 2008]; (ii) ℓ_2-regularized logistic regression (logistic) [Fan *et al.*, 2008], and (iii) ranking SVM with the linear kernel (rank-SVM) [Menon *et al.*, 2012], which is the state-of-the-art that combines ranking with isotonic regression. Note that all three classifiers are linear models with ℓ_2-regularization,

Table 1: Result obtained by the individual classifiers and various combination methods. The best and comparable results (according to the pairwise t-test with 95% confidence) are highlighted.

	method	*ijcnn1*	*covertype*	*real-sim*	*news20*	*rcv1*
MSE	SVM-lin	0.0544±0.0033	0.1798±0.0036	0.0566±0.0035	**0.0864±0.0022**	0.0495±0.0030
	logistic	0.0548±0.0023	0.1792±0.0028	**0.0544±0.0030**	0.0895±0.0023	0.0493±0.0024
	rank-SVM	0.0549±0.0030	0.1787±0.0030	0.0552±0.0026	**0.0864±0.0023**	0.0487±0.0024
	avg	**0.0522±0.0019**	**0.1774±0.0027**	**0.0544±0.0030**	0.0870±0.0022	0.0483±0.0024
	wavg	**0.0522±0.0019**	**0.1774±0.0027**	**0.0544±0.0030**	0.0870±0.0022	**0.0483±0.0024**
	MIC	0.0522±0.0020	0.1781±0.0030	0.0546±0.0029	0.0872±0.0022	0.0485±0.0022
	MR-MIC	**0.0519±0.0020**	**0.1777±0.0030**	**0.0543±0.0028**	0.0869±0.0022	**0.0483±0.0023**
AUC	SVM-lin	0.8921±0.0165	0.8037±0.0054	0.9729±0.0032	0.9504±0.0025	0.9799±0.0027
	logistic	0.9018±0.0074	0.8071±0.0054	0.9750±0.0026	0.9472±0.0028	0.9804±0.0018
	rank-SVM	0.9163±0.0042	0.8079±0.0056	0.9744±0.0018	0.9504±0.0024	0.9809±0.0021
	avg	0.9194±0.0061	0.8105±0.0045	0.9757±0.0022	0.9500±0.0024	0.9815±0.0020
	wavg	0.9196±0.0060	0.8105±0.0045	0.9758±0.0022	0.9500±0.0024	0.9815±0.0020
	MIC	**0.9204±0.0063**	**0.8116±0.0045**	0.9753±0.0021	0.9504±0.0024	0.9815±0.0023
	MR-MIC	**0.9218±0.0066**	**0.8116±0.0047**	**0.9765±0.0022**	**0.9509±0.0025**	**0.9821±0.0020**

Table 2: Results on combining dissimilar classifiers.

	method	*ijcnn1*	*covertype*	*real-sim*	*news20*	*rcv1*
MSE	SVM-rbf	0.0309±0.0014	0.1764±0.0064	0.0559±0.0036	0.0869±0.0029	0.0488±0.0026
	rank-SVM	0.0549±0.0030	0.1787±0.0030	0.0552±0.0026	0.0864±0.0023	0.0487±0.0024
	forest	0.0305±0.0012	0.1596±0.0025	0.0887±0.0022	0.1356±0.0085	0.0540±0.0039
	boosting	0.0288±0.0011	0.1602±0.0025	0.0724±0.0029	0.1119±0.0233	0.0510±0.0023
	avg	0.0266±0.0007	0.1575±0.0024	0.0571±0.0025	0.0917±0.0060	0.0435±0.0024
	wavg	0.0257±0.0007	**0.1572±0.0024**	0.0553±0.0029	0.0889±0.0051	0.0434±0.0023
	MIC	0.0251±0.0008	0.1589±0.0022	0.0537±0.0025	0.0853±0.0041	0.0431±0.0022
	MR-MIC	**0.0247±0.0007**	**0.1576±0.0021**	**0.0534±0.0025**	**0.0850±0.0042**	**0.0427±0.0023**
AUC	SVM-rbf	0.9478±0.0158	0.8104±0.0140	0.9732±0.0033	0.9498±0.0036	0.9807±0.0018
	rank-SVM	0.9163±0.0042	0.8079±0.0056	0.9744±0.0018	0.9504±0.0024	0.9809±0.0021
	forest	0.9636±0.0079	0.8463±0.0041	0.9339±0.0061	0.8885±0.0128	0.9788±0.0034
	boosting	0.9688±0.0062	0.8442±0.0046	0.9586±0.0025	0.9187±0.0339	0.9802±0.0019
	avg	0.9777±0.0073	0.8513±0.0032	0.9749±0.0020	0.9463±0.0060	0.9860±0.0016
	wavg	**0.9784±0.0071**	**0.8518±0.0033**	0.9762±0.0022	0.9488±0.0052	0.9860±0.0016
	MIC	0.9714±0.0076	0.8506±0.0039	0.9748±0.0020	0.9518±0.0041	0.9854±0.0016
	MR-MIC	**0.9791±0.0074**	**0.8513±0.0034**	**0.9774±0.0020**	**0.9528±0.0045**	**0.9863±0.0015**

and differ mainly in the loss function. Hence, as can be seen Table 1, their performance are very similar, and combining them yields only a small performance gain. This agrees with the fact that diversity is essential in an ensemble [Tumer and Ghosh, 1996]. Nevertheless, even in this "worse-case" scenario, MR-MIC still outperforms the individual classifiers and other combination approaches.

Next, we use classifiers that are more different in nature, including (i) SVM with the RBF kernel (SVM-rbf); (ii) rank-SVM (iii) random forest (forest) [Caruana *et al.*, 2008]; and (iv) boosting of 100 decision trees [Caruana *et al.*, 2008]. Results are shown in Table 2. As can be seen, the performance differences among individual classifiers are now much larger. This diversity is more commonly encountered in practice and agrees with the results in [Caruana and Niculescu-Mizil, 2006; Caruana *et al.*, 2008]. In this case, combining the calibrated probabilities, even by simple averaging, often outperforms any single classifier. Combining using the more sophisticated MIC approach performs better than averaging, while further adding manifold information enables MR-MIC to be consistently better than all the others. While the performance improvements may sometimes appear small, note that the classifiers used are powerful. Moreover, isotonic-regression based calibration is equivalent to the ROC convex hull method, and produces the optimal isotonic-transformed classifier with respect to a number of performance scores [Fawcett and Niculescu-Mizil, 2007]. Hence, any possible improvements by combining these isotonic-transformed strong classifiers are not expected to be very drastic.

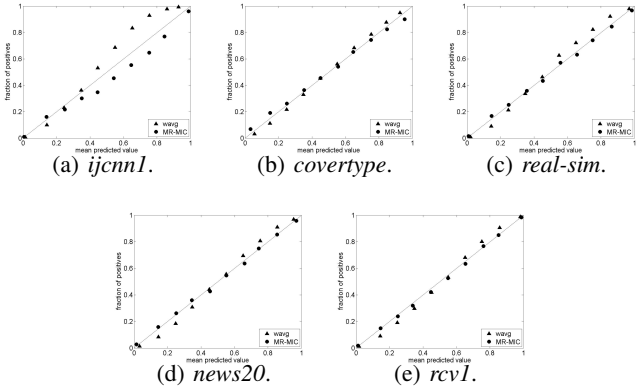
To better illustrate the relationship between performance improvement and classifier diversity, Table 3 shows the percentage MSE reduction of MR-MIC relative to the other methods. As can be seen, when the ensemble diversity is large, the corresponding improvements of MR-MIC over the

Table 3: Percentage MSE reduction of MR-MIC relative to the other methods ($\frac{\text{MSE}_{\text{method}}-\text{MSE}_{\text{MR-MIC}}}{\text{MSE}_{\text{method}}} \times 100$). The top row shows the ensemble diversity, measured by the normalized standard deviation of the base classifiers' MSE ($\frac{\text{std(MSE)}}{\text{mean(MSE)}}$). Cases where ensemble diversity is large are in bold.

	combining similar classifiers						combining dissimilar classifiers				
	ijcnn1	*covertype*	*real-sim*	*news20*	*rcv1*		*ijcnn1*	*covertype*	*real-sim*	*news20*	*rcv1*
nstd(MSE)	0.03	0.01	0.02	0.02	0.02	nstd(MSE)	**0.34**	0.06	**0.24**	**0.24**	0.05
avg	0.5	-0.2	0.2	0.1	-0.0	avg	**7.2**	-0.1	**6.4**	**7.2**	1.7
wavg	0.5	-0.2	0.1	0.1	-0.1	wavg	**3.7**	-0.3	**3.3**	**4.3**	1.5

averaging methods (avg and wavg) are also more substantial.

Figure 3 shows the reliability diagrams [Niculescu-Mizil and Caruana, 2005] for MR-MIC and its closest competitor "wavg". On 4 of the 5 data sets, points for MR-MIC lie closer to the diagonal line than those of wavg.



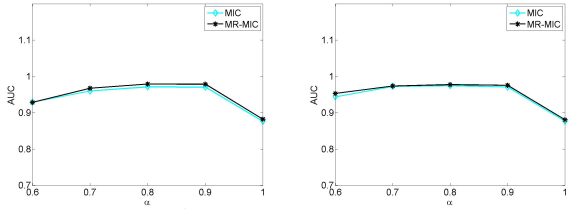
(a) *ijcnn1*. (b) *covertype*. (c) *real-sim*.

(d) *news20*. (e) *rcv1*.

Figure 3: Reliability diagrams of wavg and MR-MIC.

4.2 Variation with the Threshold α

In this section, we study the performance variation with $\alpha \in (0.5, 1]$, which is used in constructing the isotonic constraints (Section 3.1). As expected, a larger α suggests wider consensus among classifiers, and the isotonic constraints are more reliable but fewer. Experiments are performed on the *ijcnn1* and *real-sim* data sets. As can be seen from Figure 4, the performance remains relatively constant for $\alpha \in [0.7, 0.9]$. The trends on the other data sets are similar.



(a) *ijcnn1*. (b) *real-sim*.

Figure 4: Variation of the AUC with threshold α.

4.3 Manifold Regularization

Finally, we demonstrate that manifold regularization is also useful in the calibration of individual classifiers. The boosted version of 100 decision trees is used as classifier, with varying numbers of calibration samples. Results are shown in Table 4. As can be seen, manifold regularization is always useful, particularly when the amount of calibration data is limited. Figure 1 shows the corresponding isotonic regression outputs obtained. As can be seen, the use of manifold regularization leads to much smoother regression outputs.

Table 4: AUC values of the boosted trees.

data set	w/ manifold regularizer	number of calibration samples 50	200	500	1000
ijcnn1	no	0.9347	0.9540	0.9659	0.9688
	yes	**0.9732**	**0.9700**	**0.9727**	**0.9727**
covertype	no	0.8329	0.8404	0.8431	0.8442
	yes	**0.8429**	**0.8447**	**0.8455**	**0.8455**
real-sim	no	0.9408	0.9526	0.9582	0.9586
	yes	**0.9589**	**0.9590**	**0.9601**	**0.9602**
news20	no	0.9078	0.9151	0.9173	0.9187
	yes	**0.9174**	**0.9186**	**0.9192**	**0.9199**
rcv1	no	0.9662	0.9777	0.9795	0.9802
	yes	**0.9812**	**0.9811**	**0.9811**	**0.9811**

5 Conclusion

In this paper, we proposed a novel probability calibration approach by combining the prediction scores from a set of classifiers. Manifold regularization is used to avoid overfitting and ensure smoothness of the regression output over the score manifold. The extended isotonic regression model can be solved efficiently by a novel solver based on the ADMM. Experiments on a number of real-world data sets demonstrate that the proposed method consistently outperforms independent classifiers and other combinations of the classifiers' probabilities. The improvement is particularly prominent when the diversity among classifiers is large.

Acknowledgment

This research was supported in part by the Research Grants Council of the Hong Kong Special Administrative Region.

References

[Barlow *et al.*, 1972] R.E. Barlow, D.J. Bartholomew, J.M. Bremner, and H.D. Brunk. *Statistical Inference Under Order Restrictions*. Wiley, New York, 1972.

[Belkin *et al.*, 2006] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

[Boyd, 2010] S. Boyd. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2010.

[Brier, 1950] G.W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78:1–3, 1950.

[Caruana and Niculescu-Mizil, 2006] R. Caruana and A. Niculescu-Mizil. An empirical evaluation of supervised learning algorithms. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 161–168, Pittsburgh, PA, USA, June 2006.

[Caruana *et al.*, 2008] R. Caruana, N. Karampatziakis, and A. Yessenalina. An empirical evaluation of supervised learning in high dimensions. In *Proceedings of the 25th International Conference on Machine Learning*, pages 96–103, Helsinki, Finland, July 2008.

[Cormen *et al.*, 2009] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 3rd edition, 2009.

[Deng and Yin, 2012] W. Deng and W. Yin. On the global and linear convergence of the generalized alternating direction method of multipliers. Technical Report TR12-14, Rice University, 2012.

[Fan *et al.*, 2008] R.E. Fan, K.W. Chang, C.J. Hsieh, X.R. Wang, and C.J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[Fawcett and Niculescu-Mizil, 2007] T. Fawcett and A. Niculescu-Mizil. PAV and the ROC convex hull. *Machine Learning*, 68(1):97–106, July 2007.

[Fawcett, 2006] T. Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27:861–874, 2006.

[Friedman and Tibshirani, 1984] J. Friedman and R. Tibshirani. The monotone smoothing of scatterplots. *Technometrics*, 26(3):243–250, 1984.

[Gail *et al.*, 1989] M.H. Gail, L.A. Brinton, D.P. Byar, D.K. Corle, S.B. Green, C. Schairer, and J.J. Mulvihill. Projecting individualized probabilities of developing breast cancer for white females who are being examined annually. *Journal of the National Cancer Institute*, 81(24):1879–1886, 1989.

[Glowinski and Marrocco, 1975] R. Glowinski and A. Marrocco. Sur l'approximation, par elements finis d'ordre un, et la resolution, par penalisation-dualite, d'une classe de problems de dirichlet non lineares. *Revue Francaise d'Automatique, Informatique, et Recherche Opérationelle*, 9:41–76, 1975.

[Hall and Huang, 2001] P. Hall and L.S. Huang. Nonparametric kernel regression subject to monotonicity constraints. *Annals of Statistics*, 29(3):624–647, 2001.

[Jiang *et al.*, 2011] X. Jiang, M. Osl, J. Kim, and L. Ohno-Machado. Smooth isotonic regression: A new method to calibrate predictive models. In *Proceedings of the AMIA Summits on Translational Science*, pages 16–20, San Francisco, CA, USA, 2011.

[Menon *et al.*, 2012] A. Menon, X. Jiang, S. Vembu, C. Elkan, and L. Ohno-Machado. Predicting accurate probabilities with a ranking loss. In *Proceedings of the 29th International Conference on Machine Learning*, pages 703–710, Edinburgh, Scotland, UK, June 2012.

[Niculescu-Mizil and Caruana, 2005] A. Niculescu-Mizil and R. Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 625–632, Bonn, Germany, August 2005.

[Pardalos and Xue, 1999] P.M. Pardalos and G. Xue. Algorithms for a class of isotonic regression problems. *Algorithmica*, 23(3):211–222, 1999.

[Platt, 1999] J. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In A.J. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*. MIT, 1999.

[Richardson *et al.*, 2007] M. Richardson, E. Dominowska, and R. Ragno. Predicting clicks: Estimating the click-through rate for new ads. In *Proceedings of the 16th International Conference on World Wide Web*, pages 521–529, New York, NY, USA, 2007.

[Stout, 2013] Q.F. Stout. Isotonic regression via partitioning. *Algorithmica*, 66(1):93–112, May 2013.

[Tumer and Ghosh, 1996] K. Tumer and J. Ghosh. Analysis of decision boundaries in linearly combined neural classifiers. *Pattern Recognition*, 29(2):341–348, 1996.

[Wang and Li, 2008] X. Wang and F. Li. Isotonic smoothing spline regression. *Journal of Computational and Graphical Statistics*, 17(1):21–37, 2008.

[Zadrozny and Elkan, 2001] B. Zadrozny and C. Elkan. Learning and making decisions when costs and probabilities are both unknown. In *Proceedings of the 7th International Conference on Knowledge Discovery and Data Mining*, pages 204–213, New York, NY, USA, 2001.

[Zadrozny and Elkan, 2002] B. Zadrozny and C. Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, pages 694–699, Edmonton, Alberta, Canada, 2002.

[Zhou, 2012] Z.-H. Zhou. *Ensemble Methods: Foundations and Algorithms*. Chapman & Hall, 2012.